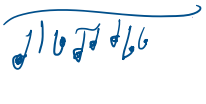
Week1 - Recurrent Neural Networks

25 June 2020 13:51

Why Sequence Models ?

- Examples of Sequence data :-

  X → Y

  1. Speech recognition :-  → "The quick brown fox jumps over the lazy dog".
  2. Music generation :- ϕ → 
  3. Sentiment classification :- "bad movie" → 1 ★
  4. DNA Sequence Analysis → AGCCCTGTACCG
  5. Machine translation → "Hermosa" → "Beautiful"
  6. Video activity recognition →  → "Running"
  7. Name entity recognition → "Yesterday Harry potter met Hermoine granger."

NOTATIONS

ex.

X : "Harry Potter & Hermoine granger invented a new Spell".

Name entity recognition

Y : 1 1 0 1 1 0 0 0 0

- input is sequence of 9 words, 9 sets of features to represent these 9 words.

$x^{<t>}$, $x^{<1>}$, $x^{<2>}$ ---- ---- $x^{<9>}$ $T_x = 9$

"temporal Sequence"

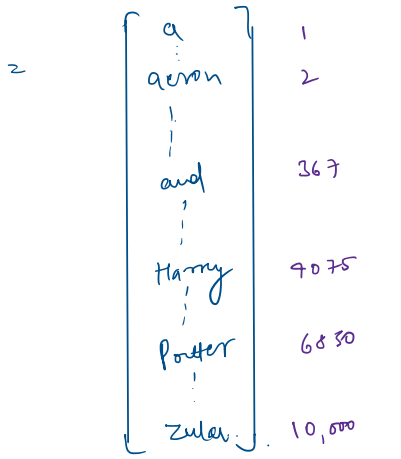
$y^{<t>}$, $y^{<1>}$, $y^{<2>}$, ---- $y^{<9>}$ $T_y = 9$

$x^{(i)<t>}$, $T_x^{(i)}$

$x^{(i)<T>}$, $Tx^{(i)}$

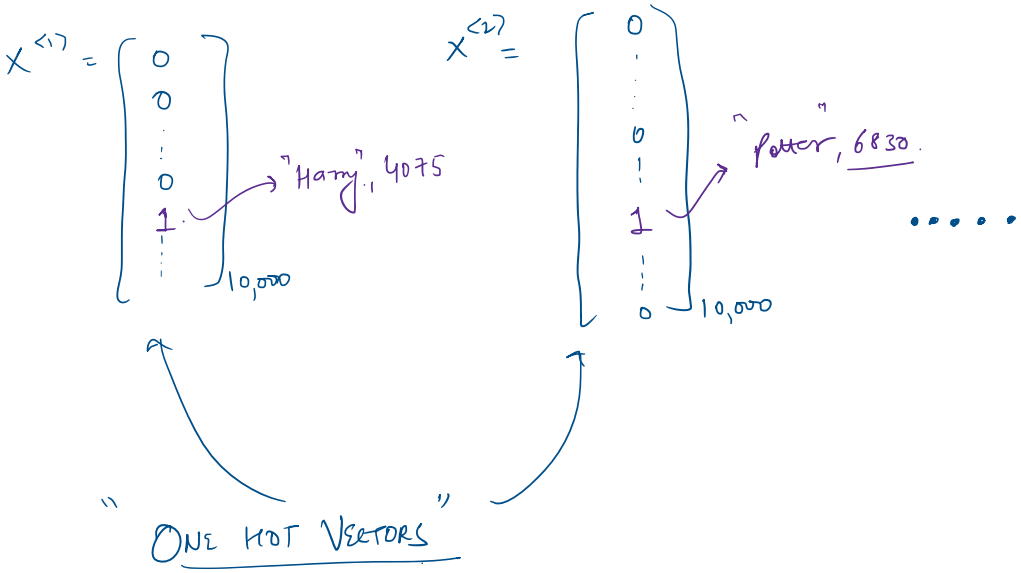
$y^{(i)<T>}$, $Ty^{(i)}$

Vocabulary / dictionary



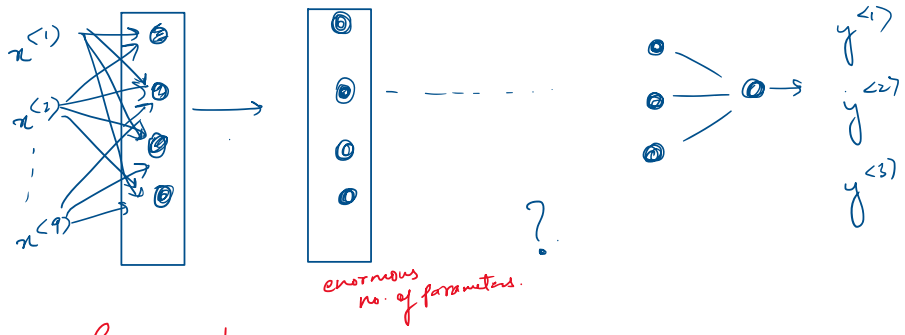
(Assuming).

One Hot representations



RECURRENT NEURAL N/W MODEL.

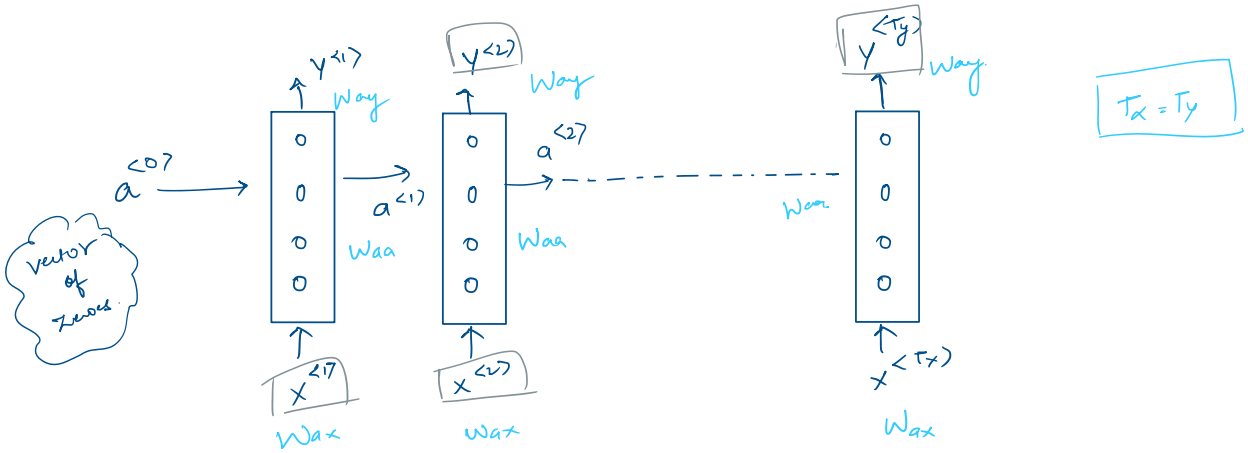
Why not a Standard Neural N/w ?



Problems!

1. I/p and O/p can be different lengths in diff. example.
2. doesn't share features learned across different position of text.

Recurrent Neural N/W

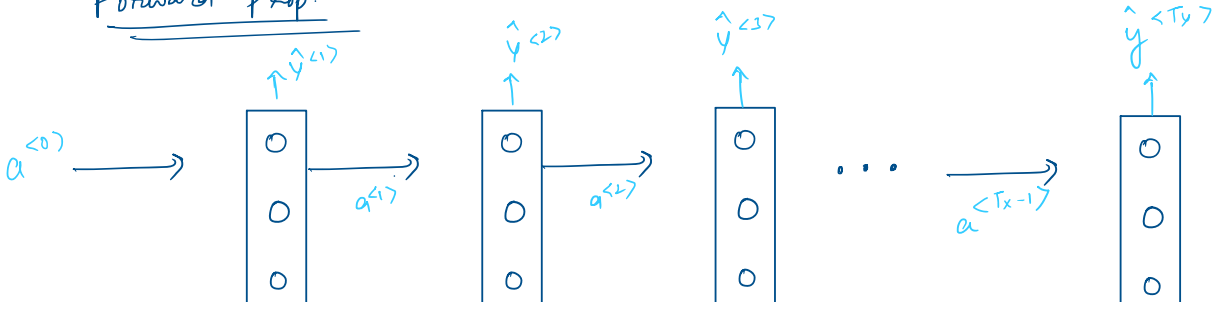


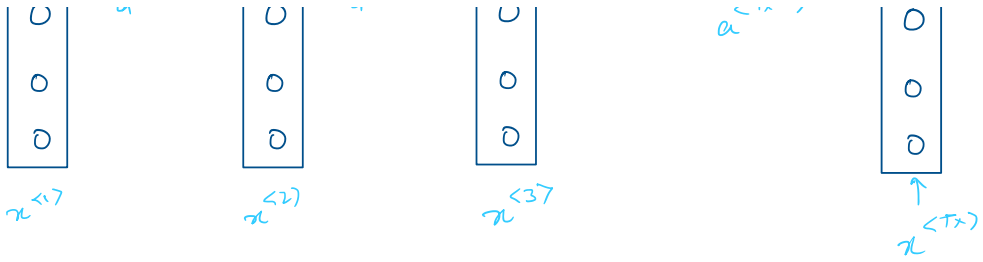
> While making prediction $y^{<2>}$, it also uses $x^{<2>}$ and $x^{<1>}$ or (earlier in the sequence.)

ex. He said "Teddy Roosevelt was a great president." ✓

He said "Teddy Bears are on Sale" ✗

Forward Prop.





$a^{<0>} = \vec{0}$

$$a^{<1>} = g_1(W_{aa} \cdot a^{<0>} + W_{ax} \cdot x^{<1>} + b_a) \quad \rightarrow \tanh(tw) \mid ReLu$$

$$\hat{y}^{<1>} = g_2(W_{ya} \cdot a^{<1>} + b_y) \quad \rightarrow \text{Sigmoid}$$

$$a^{<t>} = g(W_{aa} \cdot a^{<t-1>} + W_{ax} \cdot x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g(W_{ya} \cdot a^{<t>} + b_y)$$

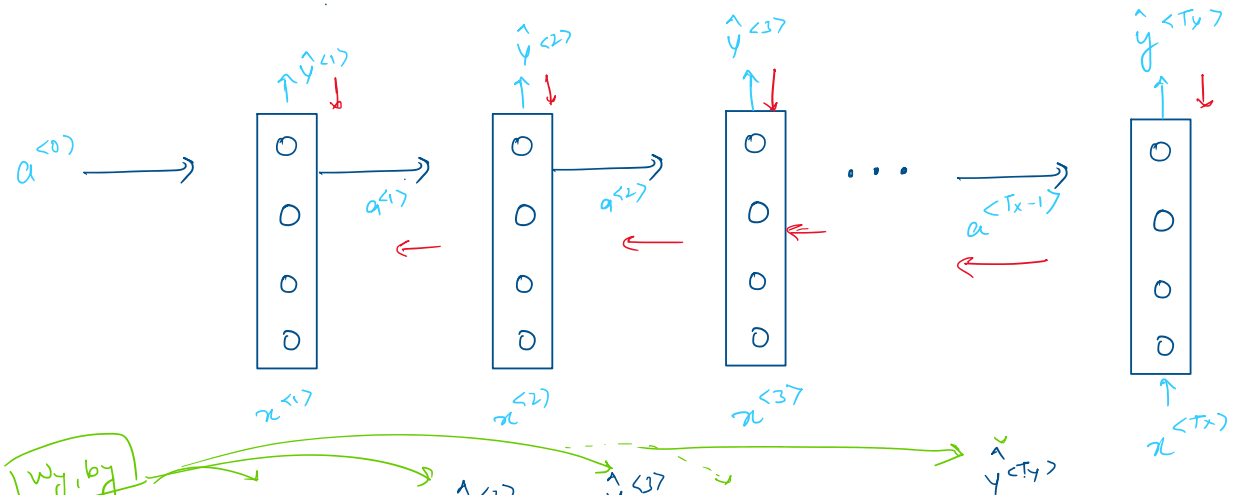
Simplified RNN Notation

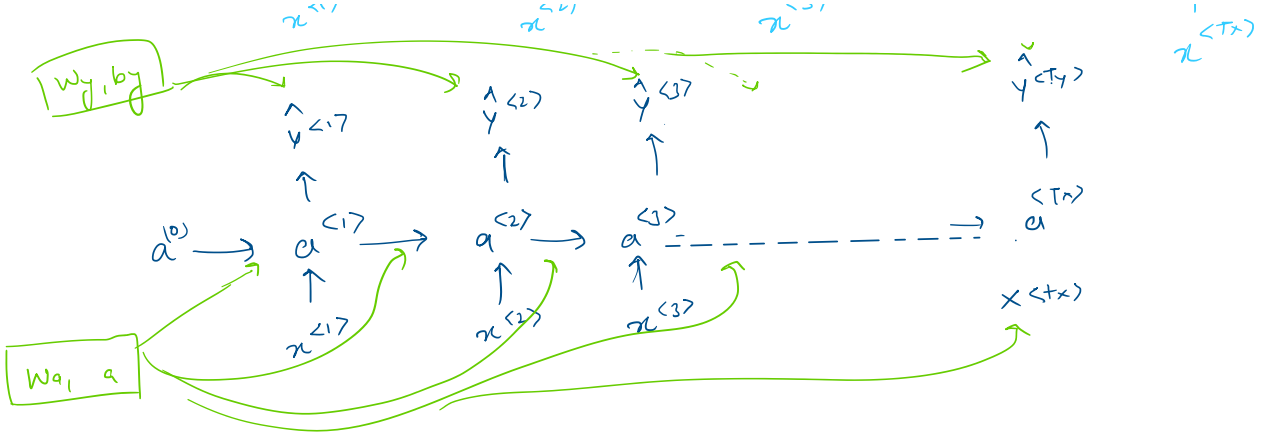
$$a^{<t>} = g(W_{aa} \cdot a^{<t-1>} + W_{ax} \cdot x^{<t>} + b_a) \iff g(W_a(a^{<t-1>} + x^{<t>}) + b_a)$$

$$\hat{y}^{<t>} = g(W_{ya} \cdot a^{<t>} + b_y)$$

$$[W_{aa} \mid W_{ax}] \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix} = W_{aa}\, a^{<t-1>} + W_{ax}\, x^{<t>}$$

Backprop. Through Time



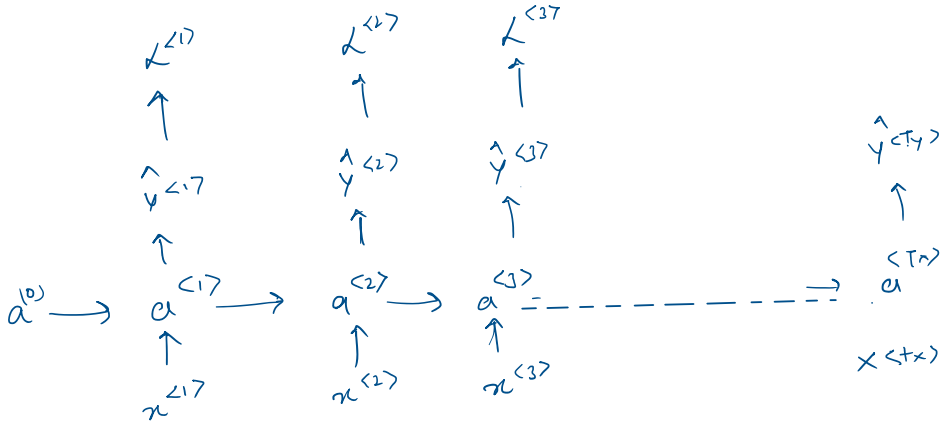


Loss fx.

$$\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{<t>}\log\hat{y}^{<t>} - (1-y^{<t>})\log(1-\hat{y}^{<t>})$$

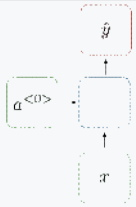
"Cross entropy loss"

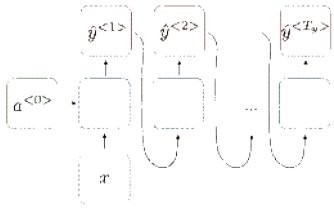
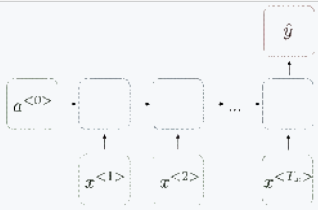
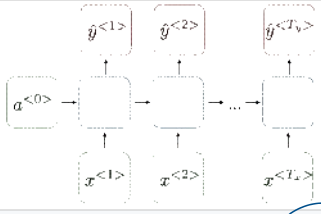
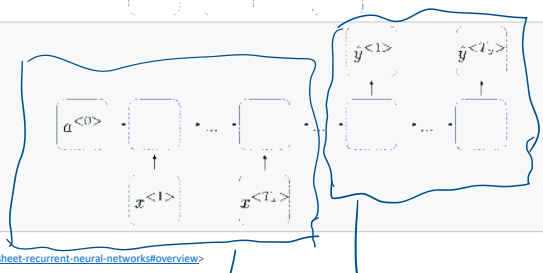
$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{Tx} \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$



Different Types Of RNN.

when $Tx \neq Ty$.

Type of RNN	Illustration	Example
One-to-one Tx = Ty = 1		Traditional neural network

One-to-many Tx = 1, Ty > 0		Music generation
Many-to-one Tx >1 , Ty=1		Sentiment classification
Many-to-many Tx = Ty		Name entity recognition
Many-to-many Tx != Ty		Machine translation

From <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks#overview>

Stanford University CS230

"Encoder"

"Decoder"

Language Model And Sequence Generation.

Speech Recognition

"The apple and pear salad."

pear / pair. ?

- P(The apple and pear salad) = 5.7×10^{-10} ✓

P(The apple and pair salad) = 3.2×10^{-13}.

P(Sentence) = ?

$y^{<1>}, y^{<2>}, y^{<3>}, \ldots\ldots\ldots y^{<T_y>}$

Language Building with RNN.

Training set :- Large Corpus of english text.

collection

Example Sentence :-

Cats average 15 hours of sleep a day.

① Tokenize (Create Vocabulary / dictionary).

$y^{<1>}, y^{<2>}, y^{<3>} \ldots\ldots\ldots y^{<T_y>}$ <EOS>

append an extra token at the end of sequence <EOS>

end of sentence.

"The Egyptian Mau is a breed of cat."

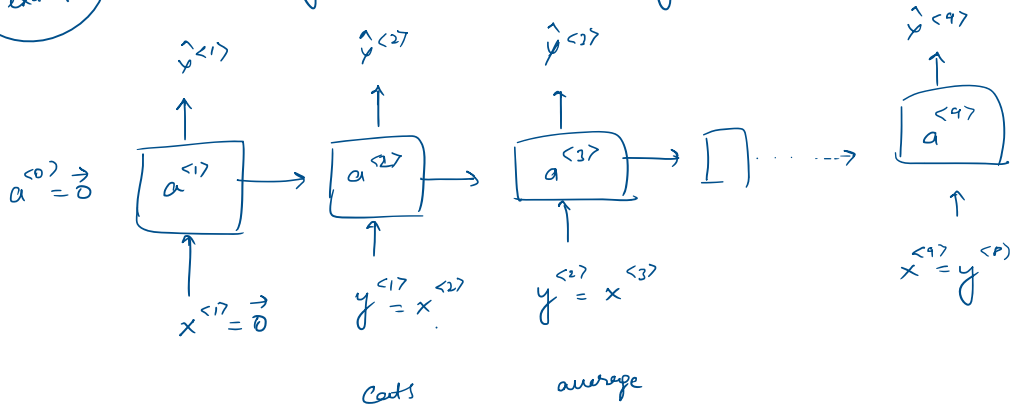
if the word is not in vocabulary replace with '<UNK>' token.

unknown word.

RNN Model

classic example

• Cats average 15 hours of sleep a day. <EOS>



$$\mathcal{L}(\hat{y}^{<t>}, y^{<t>}) = -\sum_i y_i^{<t>} \log \hat{y}_i^{<t>}$$

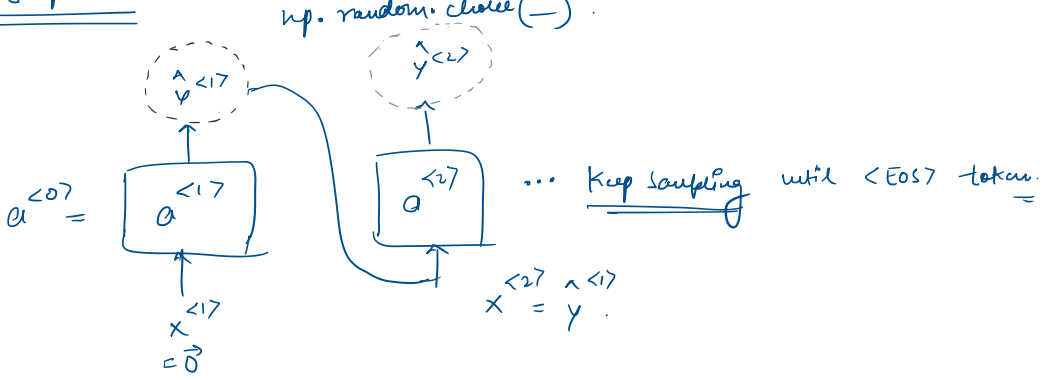
$$\mathcal{L} = \sum_t \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

SAMPLING NOVEL SEQUENCES.

$P(y^{<1>}, y^{<2>}, y^{<3>} \dots y^{<T_x>})$

Sample

np.random.choice(—)



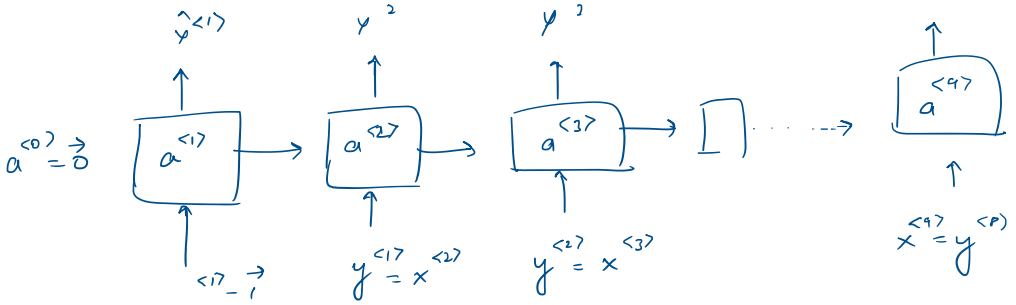
... Keep sampling until <EOS> token.

Character level vocab. —

Vocabulary = [a, b, c . . . z, A-Z, ., ,, ;, :, "- - - -].

Very expensive.

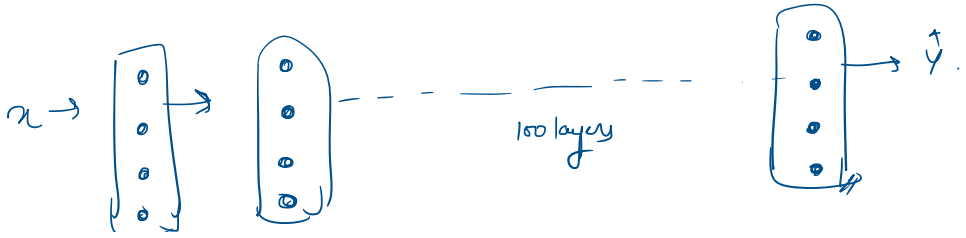
VANISHING GRADIENTS WITH RNNS



Sentence : "The cat which already ate - - - - - - - - , was full."

Cat is singular so
Cat : was
Cats : were

ex. in deep neural nfw :→ (100 layers)



forward prop. →

← Backprop.

(*) if the layer is very deep, we will have hard time calculating gradient, affecting weights of earlier layers.

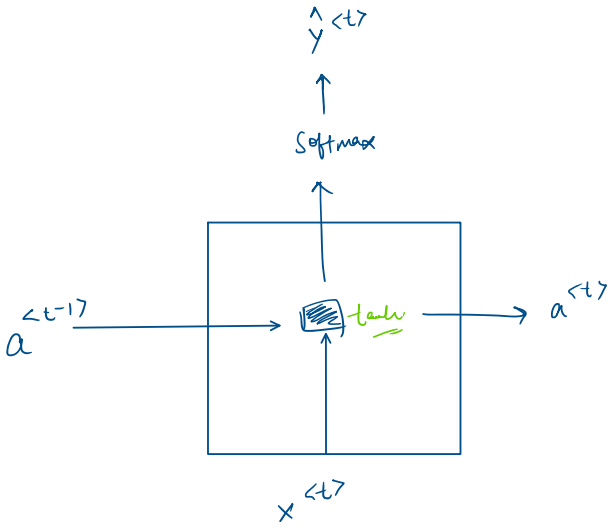
Vanishing / Exploding gradients → by tuning week 1.

→ gradient clipping

GATED RECURRENT UNIT (GRU)

tanh

$$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$$



"The cat which already ate ---, was full."

Cat : was
Cats : were

C = memory cell. } Provide memory to remember that the "cat" was singular / plural

$c^{<t>}$

C = memory cell. Provide memory to remember that the "cat was singular / plural for further consideration.

$c^{\langle t \rangle}$, at time t

$c^{\langle t \rangle} \xrightarrow{\text{o/p}} a^{\langle t \rangle}$

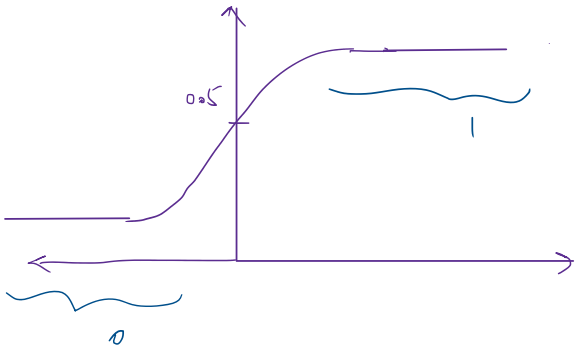
$\tilde{c}^{\langle t \rangle} = \tanh\left(W_c [c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c\right)$.

Candidate for replacing $c^{\langle t \rangle}$

update gate

$\Gamma_u = (0,1) = \sigma\left(W_u [c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c\right)$.

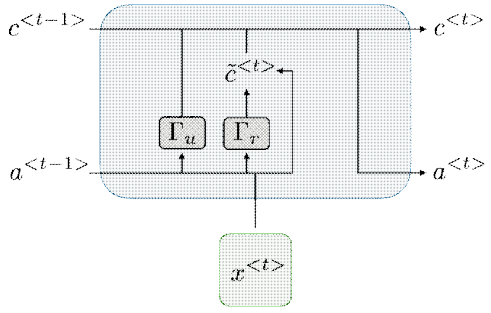
(gamma)



{ singular : 1
plural : 0 }.

$$c^{\langle t \rangle} = \Gamma_u * \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$$

FULL GRU



h $c^{\langle t \rangle} = \gamma_u * \tilde{c}^{\langle t \rangle} + (1-\gamma_u) * c^{\langle t-1 \rangle}$

"relevance"

$\tilde{h}$ $\tilde{c}^{\langle t \rangle} = \tanh\left(W_c\left[\gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$

r $\gamma_r = \sigma\left(W_r\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_r\right)$

u $\gamma_u = (0,1) = \sigma\left(W_u\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$.

$$\boxed{a^{\langle t \rangle} = c^{\langle t \rangle}}$$

LONG-SHORT TERM MEMORY. (LSTM)

$$c^{\langle t \rangle} = \gamma_u * \tilde{c}^{\langle t \rangle} + (1-\gamma_u) * c^{\langle t-1 \rangle}$$

$$\tilde{c}^{\langle t \rangle} = \tanh\left(W_c\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right).$$

$$\gamma_r = \sigma\left(W_r\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_r\right)$$

$$\gamma_u = (0,1) = \sigma\left(W_u\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right).$$

$$\gamma_f = \sigma\left(W_f\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_f\right)$$

"forget gate"

$$\gamma_o = \sigma\left(W_o\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_o\right)$$

"output"

$$c^{\langle t \rangle} = \left[\gamma_u * \tilde{c}^{\langle t \rangle}\right] + \left[\gamma_f * c^{\langle t-1 \rangle}\right]$$

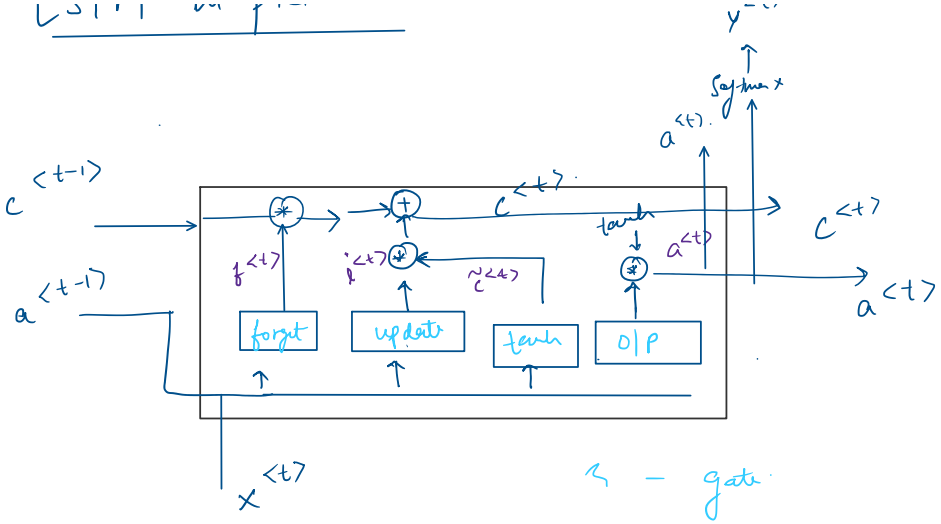
$$\boxed{a^{\langle t \rangle} = \gamma_o * \tanh\left(c^{\langle t \rangle}\right).}$$

LSTM in picture

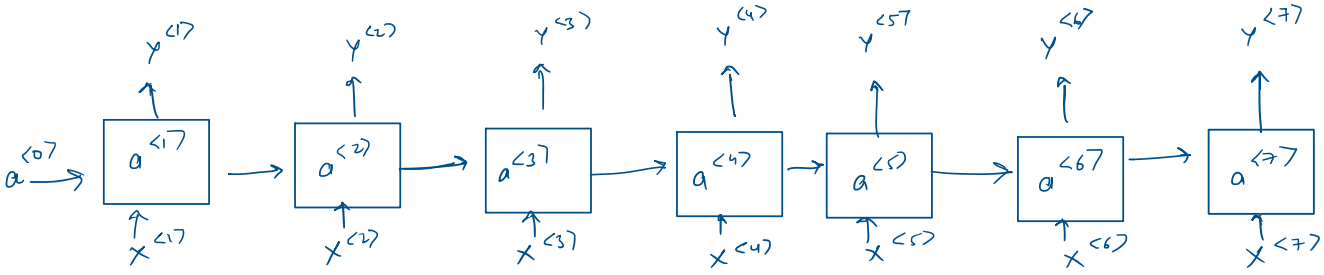
$y^{\langle t \rangle}$

↑

Softmax



BIDIRECTIONAL RNN. (BRNN)

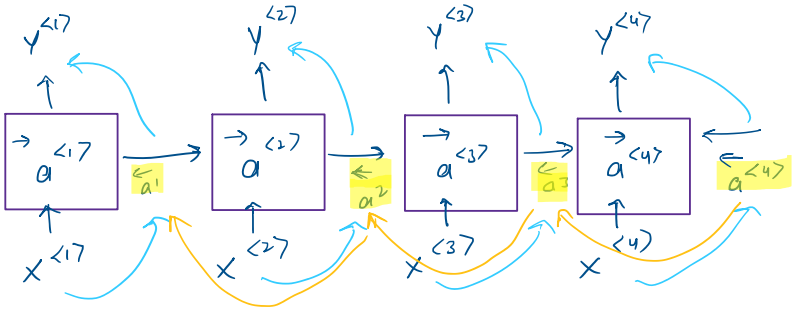


He said "Teddy Bears are on sale"

He said "Teddy Roosevelt was a great President"

- Unidirectional RNN.

$$\hat{y}^{<t>} = g\left(W_y\left[\vec{a}^{<t>}, \overleftarrow{a}^{<t>}\right] + b_y\right)$$

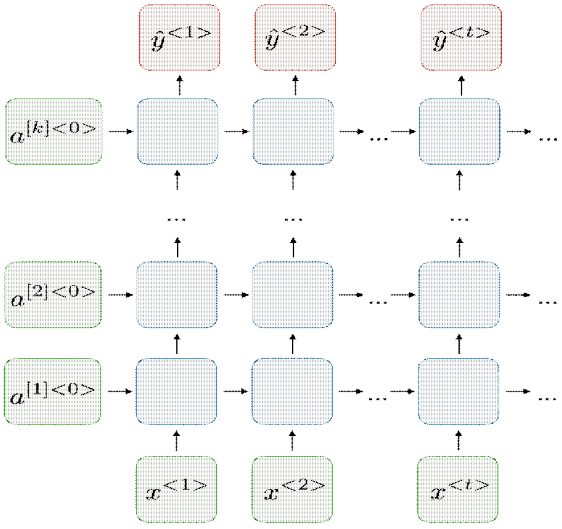


Acyclic graph

4 word sentence.

" BRNN with LSTM "

Deep RNN.



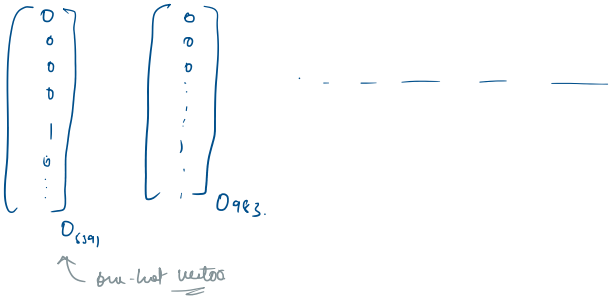
M1 Introduction To Word-Embeddings

Word Representation

$V = [a, aaron, \ldots\ldots Zulu, <UNK>]$

1 hot representation

Man	Women	King	Queen
(5391)	(9853)	(4914)	(7157)



one-hot vectors

Featured Representation

	Man	Women	King	Queen	Apple	Orange
gender	-1	1	-0.95	0.97	0.00	0.01
royalty	.01	.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	.03	-0.2
food						
Size						
⋮						

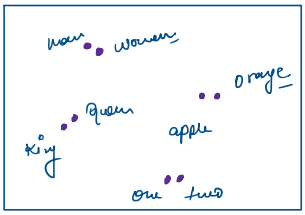
e_{5391}

"I want a glass of Orange juice"

"I want a glass of apple juice"

* Orange and apple are highly correlated ("fruit").

Visualizing Word Embeddings

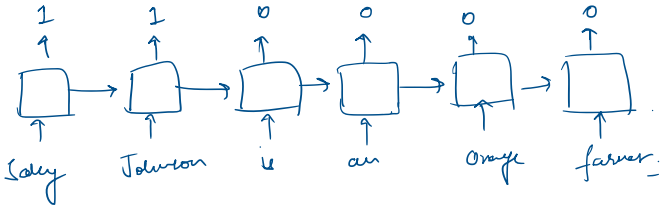


t- SNE.

USING WORD EMBEDDINGS

"Name Entity Recognition"

Sally Johnson is an Orange farmer.



"Robert is an durian farmer"

a

Cultivator.

⇒ "Cultivator & farmer are highly Correlated."

Transfer learning

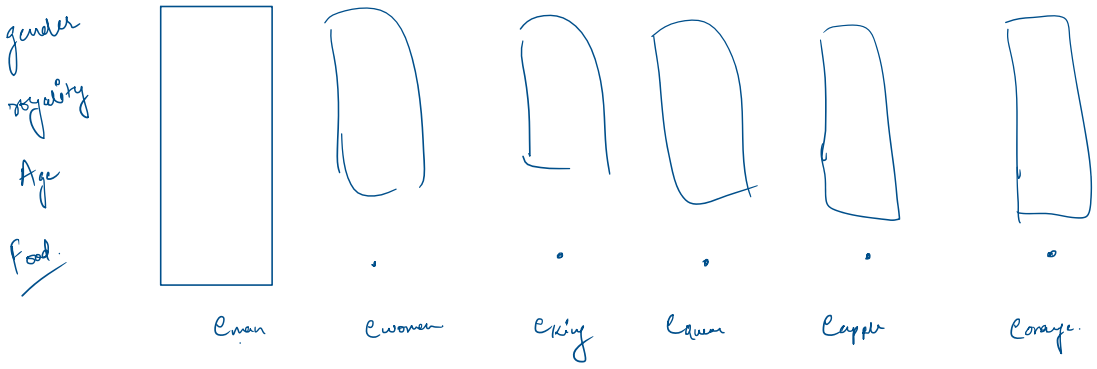
* > Learn word embedding from large text corpus.
* > Transfer embeddings to new task with smaller training set.
> Continue to fine tune the word embeddings with new data.

PROPERTIES OF WORD-EMBEDDINGS

Analogies.

Man : Women :: King : ?

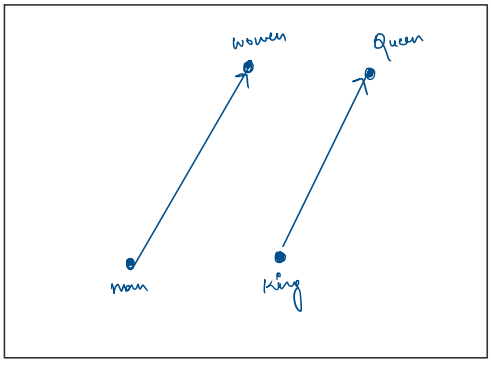
Man	Women	King	Queen	Apple	Orange
(5391)	(9853)	(4914)	(7157)	(456)	(6257)



$$e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$e_{man} - e_{woman} \approx e_{king} - [\ldots]$$

$$\Rightarrow \boxed{e_{man} - e_{woman} \approx e_{king} - e_{queen}}$$



word space

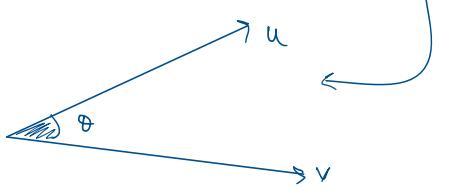
find word w, s.t.:

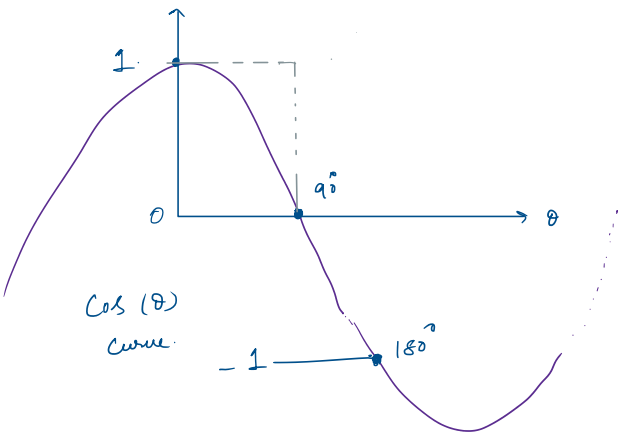
$$\max \text{Sim}(e_w, e_{king} - e_{man} + e_{woman})$$

- find a similarity fx.

$\Rightarrow$ Cosine similarity.

$$\text{sim}(u, v) = \frac{u^T v}{\|u\| \|v\|} = \cos(\theta)$$





θ	Cos(θ)
0	1
π/2	0
π	-1
2π	1

Other similarity fx:

- Euclidian distance. $\|u - v\|^2$

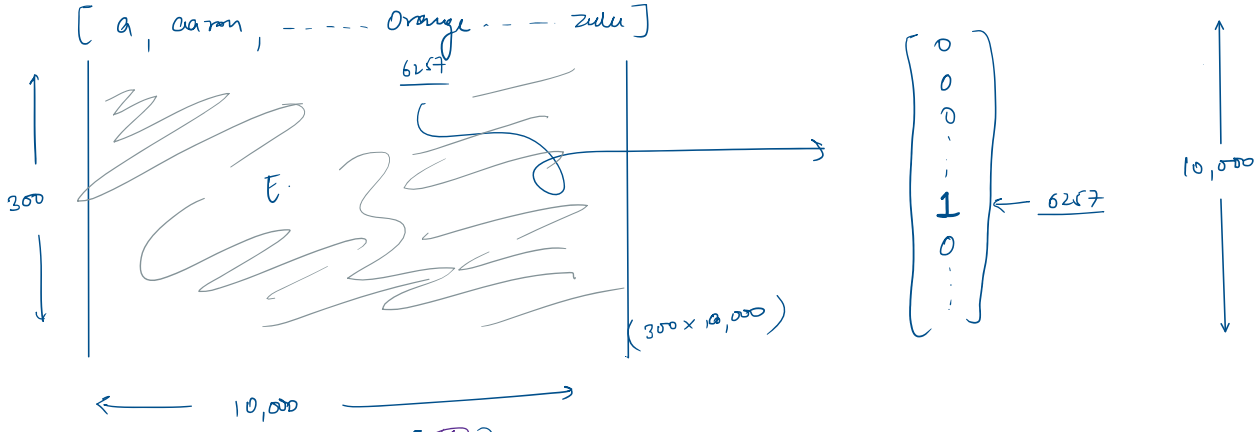
India : delhi :: Kenya : "Nairobi"

EMBEDDING MATRIX.

embedding matrix (E)

(300 x 10,000) → if you have 10,000 word dictionary

[a, aaron, ... Orange ... zulu]



$E \cdot O_{6257} = e_{6257}$

O_{6257} → one-hot vector.

$E \cdot O_j = e_j$

Embedding for word j.

Embedding for words.

* It is not efficient to use multiplication of matrix, as the cost is very high.

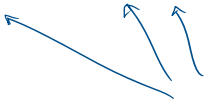
M2 LEARNING WORD EMBEDDINGS : Word2Vec and Glove.

LEARNING WORD EMBEDDING

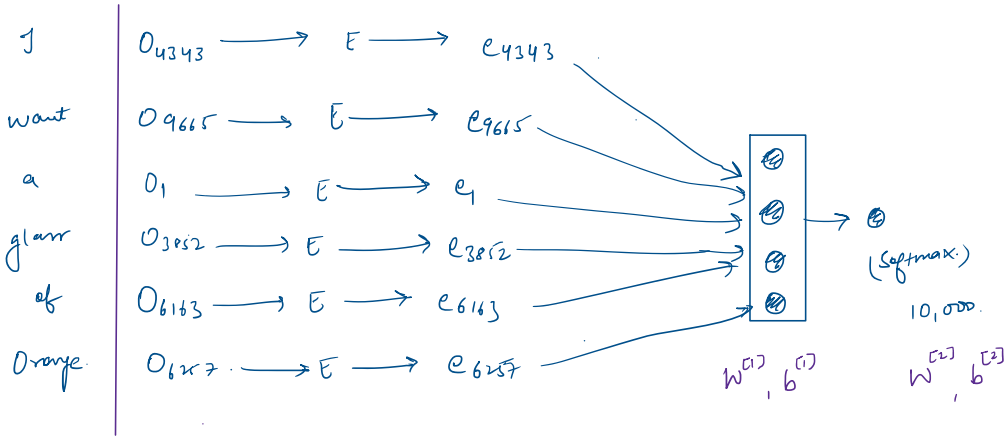
Natural language Model.

eg I want a glass of Orange _____.

(4343) (9665) (1) (3852) (6163) (6257)



indexed in vocab.



WORD 2 VEC.

Skip-grams : I want a glass of orange juice along with my cereal.

Context	Target
Orange	juice
Orange	glass
Orange	my

randomly selected

Model

Vocab - Size = 10,000 K

Context c ("Orange") (6257) → Target t ("juice") (4834)

$x \longrightarrow y$

$O_c \rightarrow E \rightarrow e_c \rightarrow$ [Softmax unit] $\rightarrow \hat{y}$

$e_c = E \cdot O_c$

Softmax : $p(t/c) = \dfrac{e^{\theta_t^T \cdot e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T \cdot e_c}}$

θ_t = parameter associated with o/p t (chance of t, being a label)

$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{1000} \left[y_i \cdot \log \hat{y}_i \right]$

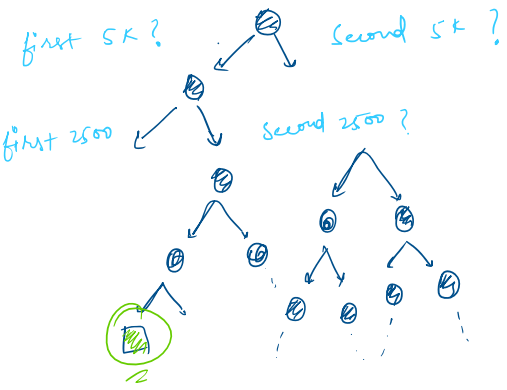
Problems with Softmax Classification.

- Computational Speed.

$p(t/c) = \dfrac{e^{\theta_t^T \cdot e_c}}{\sum_{i=1}^{10,000} e^{\theta_j^T \cdot e_c}}$

every time it will calculate 10k times, at higher values, it will compute slower.

⇒ Hierarchical Softmax.



NEGATIVE SAMPLING

predict (context, target) pair:

Context	target	Probability
Orange	juice	≈ 1
Orange	king	≈ 0
Orange	book	≈ 0

K

find p with random K target and fill from (0).

" Create a supervised learning algorithm, given (Context, target) predict probability. "

$$K = \begin{cases} 5\text{-}20, & \text{for small datasets} \\ 2\text{-}5, & \text{for larger datasets} \end{cases}$$

Softmax Model :>

$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

Context	word	target
- - - -	- - -	1
- - - -	- - -	0
- - -	- - -	0
- - -	- -	0

Use Logistic Regression Model.

$$p(y=1 \mid c,t) = \sigma(\theta_t^T e_c)$$

→ Sigmoid

"Orange" 6257

$O_{6257} \rightarrow E \rightarrow e_{6257}$ → juice? / king? / ...

1, real ex.

k, random ex.

Selecting -ve examples?

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10,000} f(w_j)^{3/4}} = \frac{1}{|V|}$$

uniform distribution raised to 3/4th.

$\sum_{i=1}^{10,000} f(w_j)^{3/4}$ $|V|$ to 5/9 ??.

GloVe Word Vectors

global vectors for word representation.

" I want a glass of Orange juice to go along with my cereal "

X_{ij} = # of times j appears in the context of i
(j = t, i = c)

minimize $\sum_{i=1}^{10K} \sum_{j=1}^{10K} f(X_{ij}) \left(\theta_i^T e_j + b_i + b_j - \log X_{ij} \right)^2$

$i, j = t, c$

or $(\theta_t)^T . e_c$

correlation

$f(X_{ij})$

this is the f at ------

(stop words)

* might give more weight to stop words and less on more attention words.

if $(X_{ij} = 0)$ then $\log 0$ = undefined, so we will add weighting term $f(X_{ij})$, $f(X_{ij}) = 0$, if $X_{ij} = 0$.
so $0 . \log 0 = 0$

f can be chosen to be heuristic to accomplish this.

Take average after calculating.

$$e_w^{(final)} = \frac{e_w + \theta_w}{2}$$

M3.

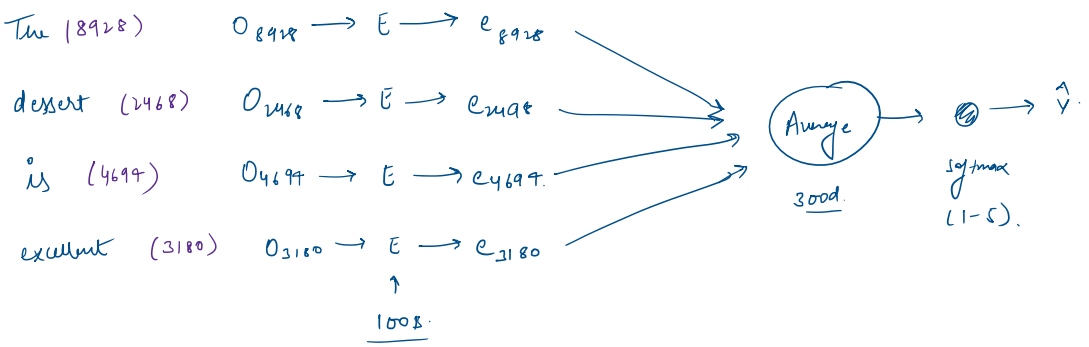
Applications Using Word Embeddings.

Sentiment Classification.

x ⟶ y

" dessert is excellent " ★★★★★

"dessert is excellent"	5 ★
"Service is slow"	1 ★
"good for quick meal, but nothing special"	3 ★

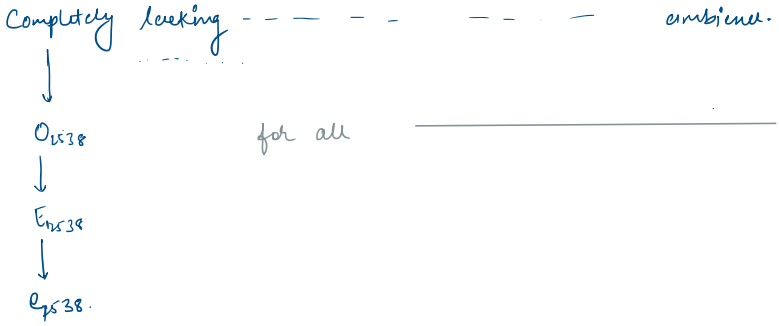


> ignores word order.

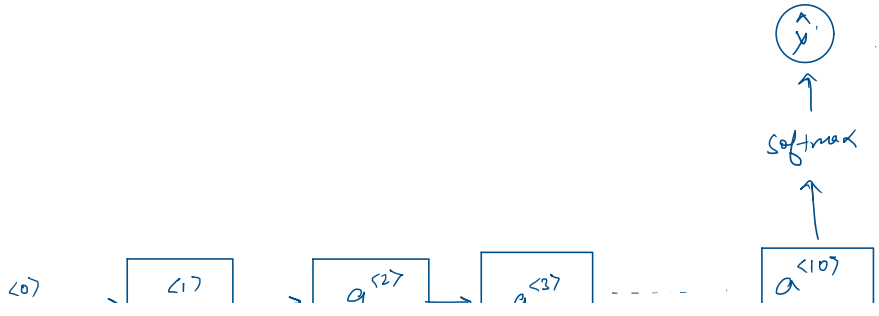
ex: "Completely lacking in good taste, good service and good ambience."

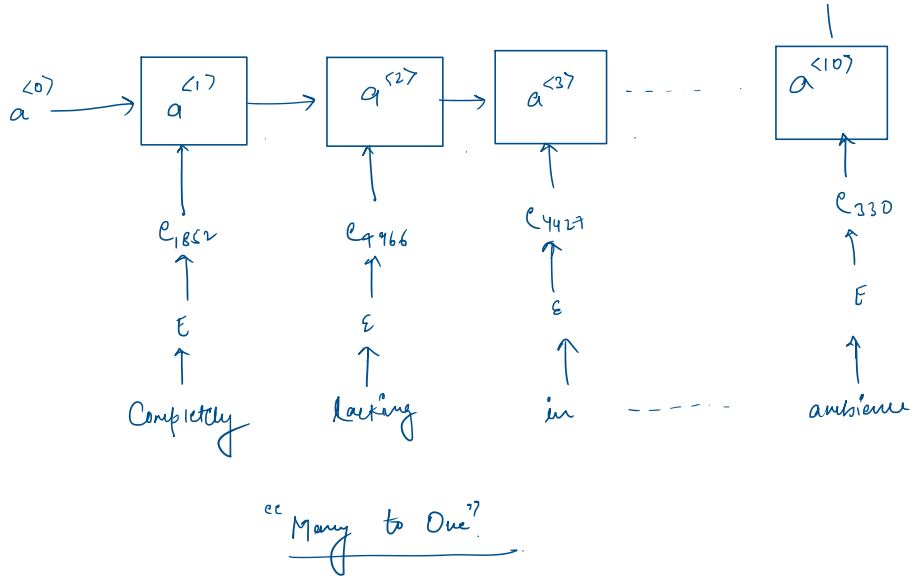
* negative review, but "good" appears a lot!

RNN for Sentiment Classification



① Calculate one-hot vector.
② Embedding Matrix
③ find embedding vector. ($O * E$).
④ Feed e to RNN





"Many to One"

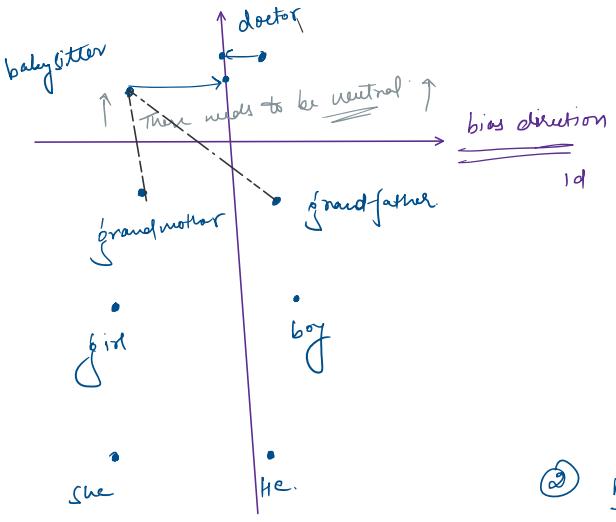
DEBIASING WORD EMBEDDINGS

Man : Women :: King : Queen

Man : programmer :: Women : Homemaker ✗

Father : doctor :: Mother : Nurse ✗

* Word embeddings can reflect gender, ethnicity, age, sexual Orientation, and other biases of text used to train the model.



① Identify bias direction.

(Taking gender bias here.)

$e_{he} - e_{she}$

$e_{male} - e_{female}$

$e_{boy} - e_{girl}$

} Average

② Neutralize: For every word that is not definitional, project to get rid of bias.

③ Equalize pairs :-

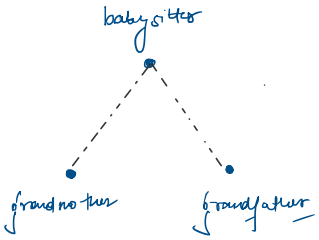
grandmother	grandfather
girl	boy

} should be at equivalent distance.

grandmother | grandfather } should be at equivalent distance.

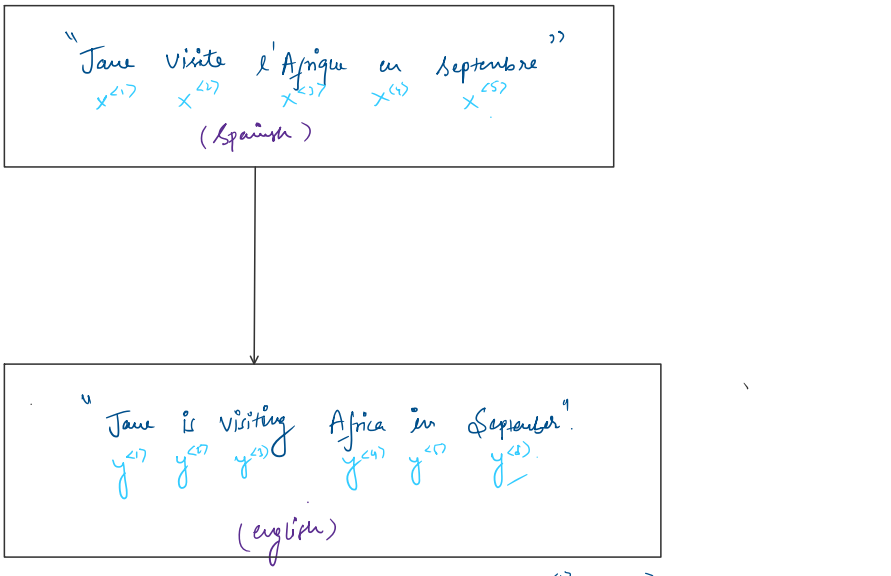
girl | boy

* Move grandmother and grandfather at pair points equidistant from a axis.



M1. Various Sequence to Sequence Architectures.

Basic Model



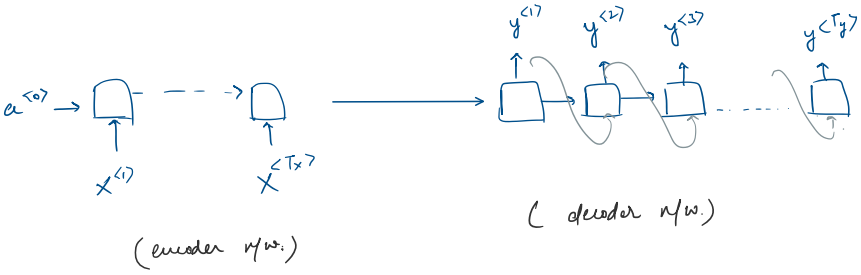
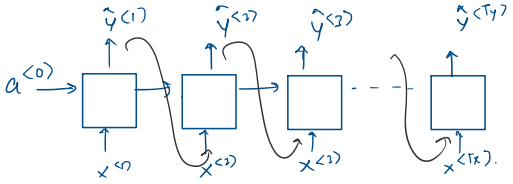


Image Captioning

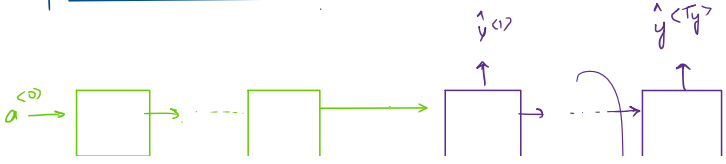
→ "A cat sitting on a chair"

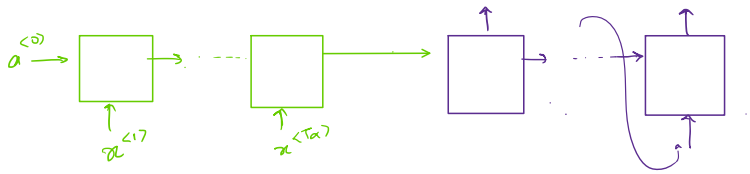
Picking the most likely sentence

Language Model (as in first week)



Machine Translation Model ⇒





encoded n/w. decoded n/w.

Conditional N/W.

" Jane visited l' Afrique en Septembre."

$P(\hat{y}^{<1>}, \hat{y}^{<2>}, \hat{y}^{<3>}, \hat{y}^{<4>} \ldots\ldots)$

→ Jane is visiting Africa in September. — good translation

→ Jane is going to be visiting Africa in September.

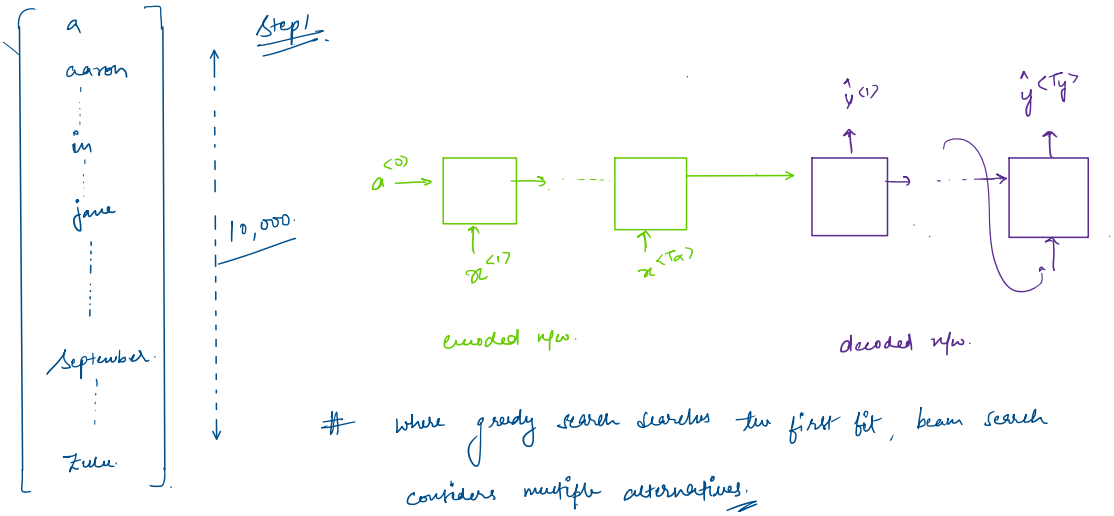
→ In September Jane will visit Africa.

→ Her African friend welcomed Jane in September. — bad translation

" beam search"

Beam Search.

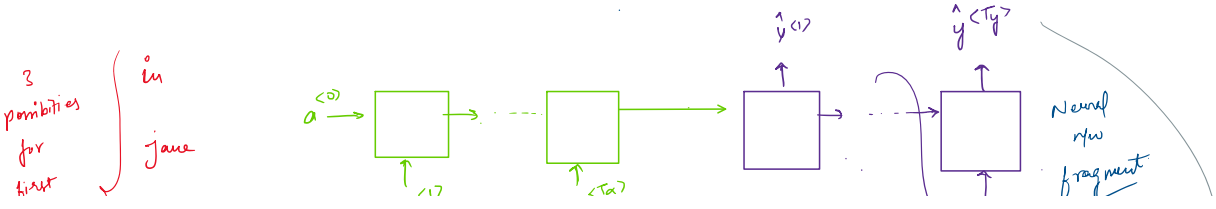
> Beam Search Algorithm

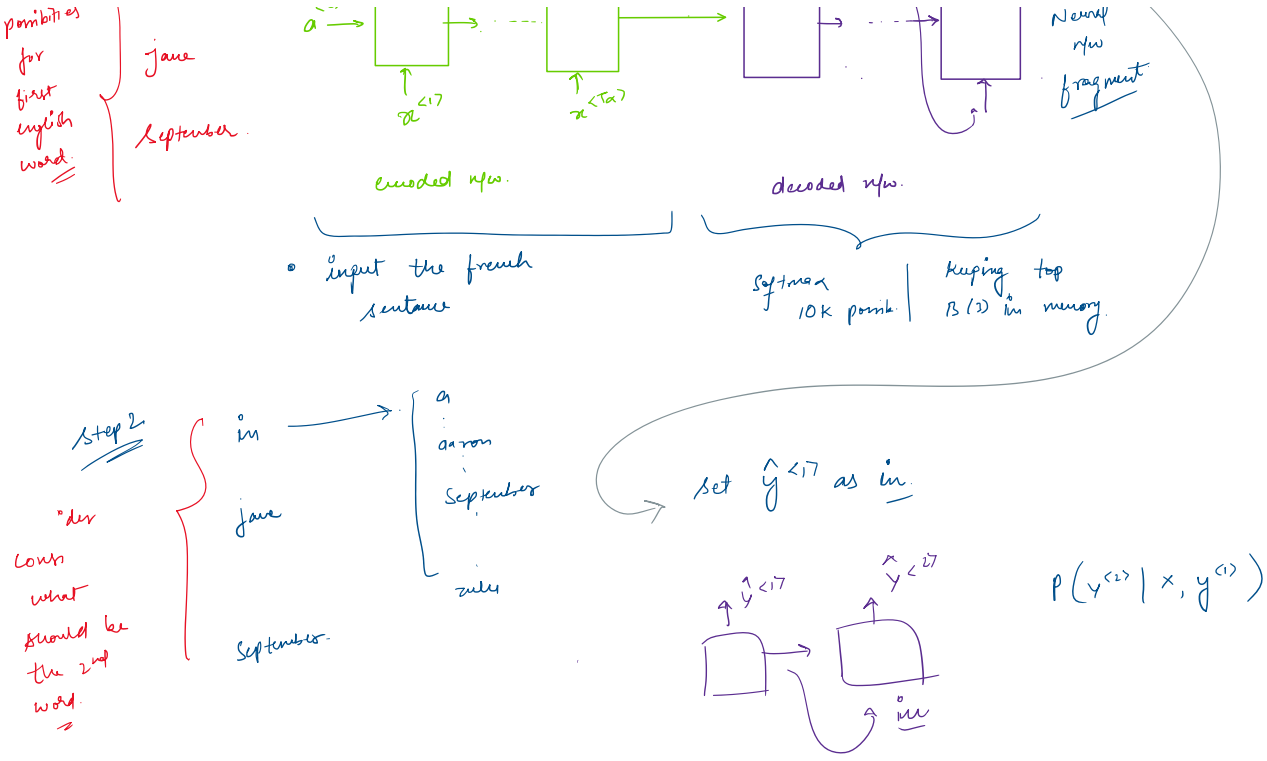


where greedy search searches the first bit, beam search considers multiple alternatives.

* beam search algorithm has a parameter B (Beam width)

B = 3 (say). • Considers 3 possibilities.





pairs of $y^{<1>}, y^{<2>}$ given x.

$$P(\hat{y}^{<1>}, y^{<2>} \mid x) = P(y^{<1>} \mid x) \cdot P(y^{<2>} \mid x, y^{<1>})$$

Conditional probability

... for all the words

Refinements to Beam Search

Length Normalization

$$\arg\max_{y} \prod_{t=1}^{T_y} P(y^{<t>} \mid x, y^{<1>}, \ldots, y^{<t-1>})$$

log

$$P(y^{<1>} \ldots y^{<T_y>} \mid x) = P(y^{<1>} \mid x) \cdot P(y^{<2>} \mid x, y^{<1>}) \cdot P(y^{<3>} \mid x, y^{<1>}, y^{<2>})$$

* Normalize by number of words.

$$\frac{1}{T_y^{\alpha}} \sum_{t=1}^{T_y} \log P(y^{<t>} \mid x, y^{<1>}, \ldots, y^{<t-1>})$$

Beam Width ?

Larger :- Higher Accuracy, more Computational.

Smaller :- Low Accuracy, Faster

Error Analysis On Beam Search.

Beam search is an approximate search algorithm / heuristic search algorithm.

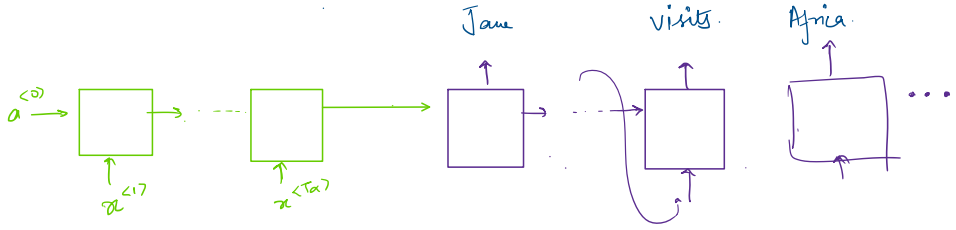
"Jane visite l'Afrique en septembre."

Human : Jane visits Africa in September. (y^*)

Algorithm : Jane visited Africa last September. $(\hat{y})$

- RNN (encoder and decoder)
- Beam Search.

RNN.



encoded n/w. decoded n/w.

max. $\left(P(y^* | x) , P(\hat{y} | x) \right)$

Jane visits Africa in September. (y^*) $P(y^* | x)$

Jane visited Africa last September. $(\hat{y})$ $P(\hat{y} | x)$

Case 1 : $P(y^*) > P(\hat{y})$

⇒ but beam search chooses $P(\hat{y})$.

⇒ Beam Search is at fault.

Case 2 : $P(y^*) < P(\hat{y})$

⇒ y^* is a better translation than $\hat{y}$. But RNN predicts

$$P(y^*) < P(\hat{y})$$

⇒ RNN model is at fault.

Bleau Score.

(Billingual Evaluation understudy Score.)

- given a french sentance that can be multiple english translation, this can be solved with "Bleau Score"

French : Le chat est sur le tapis.

Reference 1 : The cat is on the mat. { the : 2 }

Reference 2 : There is cat on the mat. { the : 1 }

BLEAU Score measure how good a machine translation is.

Precision (1) :- $\frac{2}{7}$

Blue Score on Bigrams

"pairs of words next to each other"

Reference 1 : The cat is on the mat.

Reference 2 : There is cat on the mat.

MT O/P : The cat the cat on the mat

(Machine Translation)

	MT O/P Count	Reference Countclip
the cat	2	1
Cat the	1	0
Cat on	1	1
on the	1	1
the mat	1	1

Modified bigram Precision :- $\frac{\sum \text{Count clips}}{\sum \text{Count}}$

$$= \frac{4}{6} = \frac{2}{3}$$

Bleu Details

p_n = Bleu Score of n-grams.

P_1, P_2, P_3, P_4

Combined blue Score : $BP \exp\left(\frac{1}{4}\sum_{n=1}^{4} P_n\right)$.

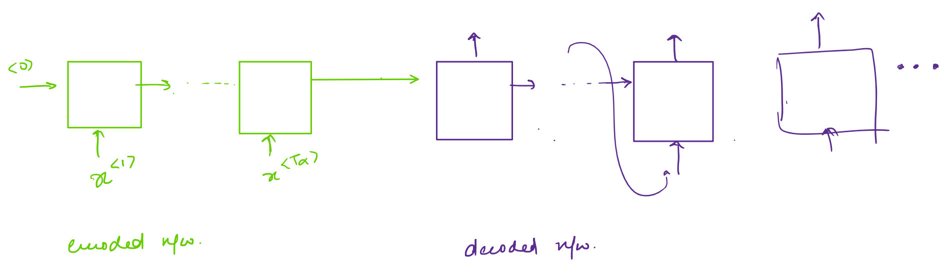
(brevity penalty)

$$BP \begin{cases} 1, & \text{if MT\_Output length} > \text{reference o/p Length} \\ \exp(1 - \text{reference\_o/p Length} / \text{MT\_o/p Length}) \end{cases}$$

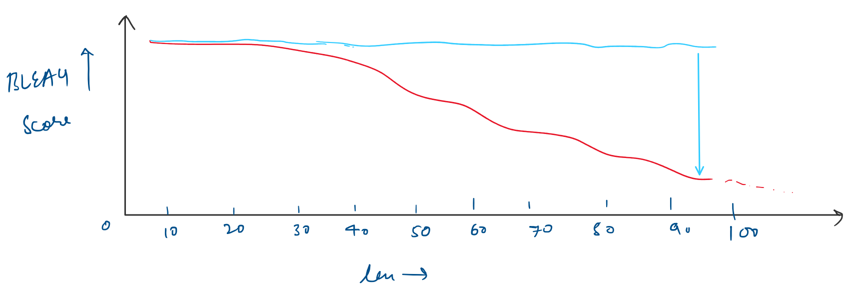
Attention Model Intuition.

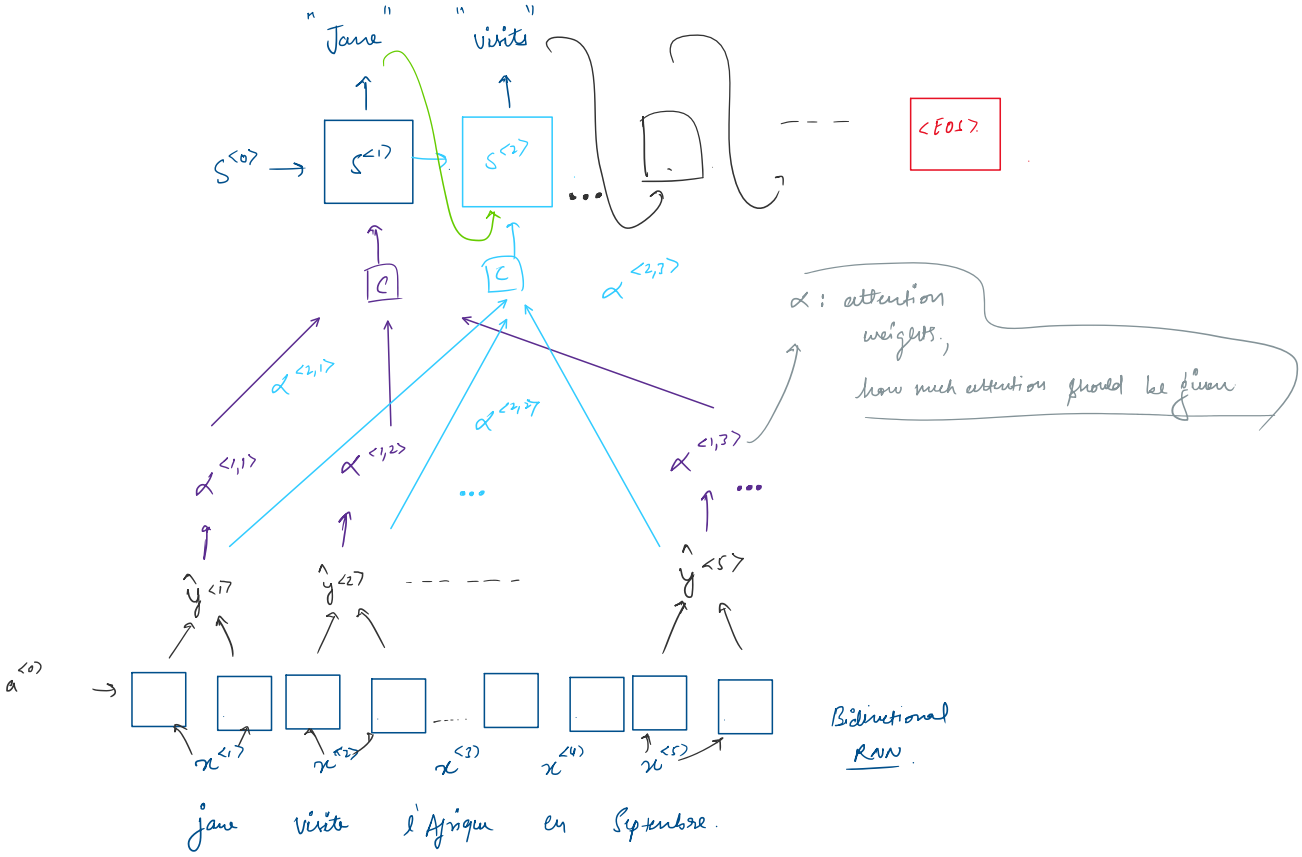
The problem of long Sequence

> given a long french sentance ____________

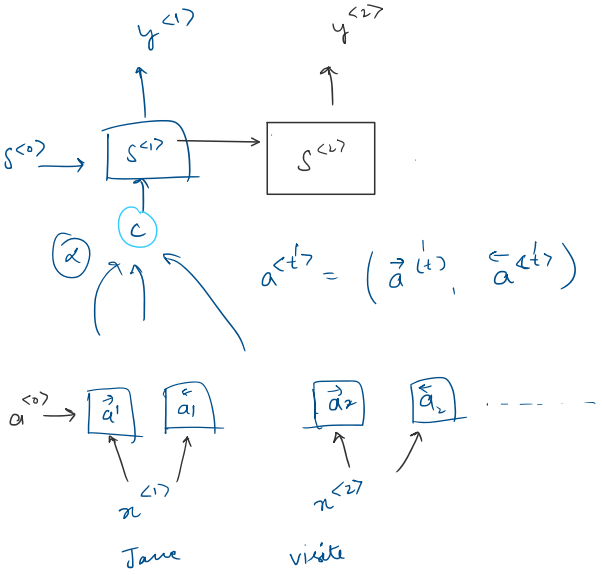


* It is hard to memorize the whole sentance, what humans will do is translate a part of sentance and move forward.





Attention Model



$$\sum_{t'} \alpha^{<1,t'>} = 1$$

$$c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$$

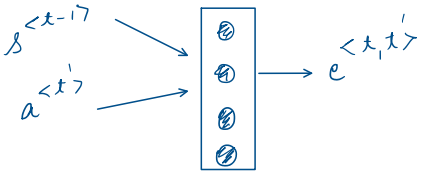
$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

Computing Attention

$a^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

$$\frac{\exp(e^{<t,t'>})}{}$$

$$\alpha^{\langle t,t' \rangle} = \frac{\exp\left(e^{\langle t,t' \rangle}\right)}{\sum_{t'=1}^{T_x} \exp\left(e^{\langle t,t' \rangle}\right)}$$



- takes quadratic cost. $(T_x)^{T_y}$

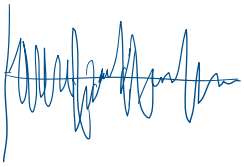
T_x = input
T_y = Output

M2 Speech Recognition

Speech recognition

$x \longrightarrow y$

(audio clip) (transcript)

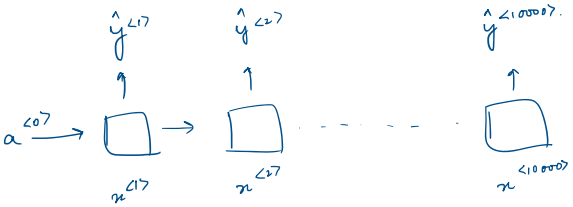


"the quick brown fox ---"

time →

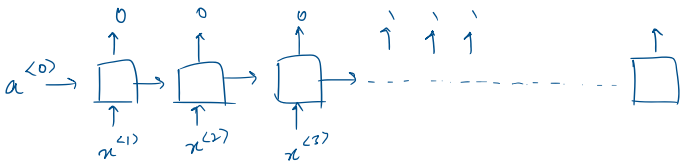
CTC Cost

"Connectionist temporal classification"



Trigger Word detection

Spectrogram



"Hey Siri !!"

"OK google !!"